Amendments to the Specification

Please amend the specification as follows:

Please amend paragraph [0005] as follows:

[0005] The present invention provides a system and method for processing database queries across network boundaries. The present invention can be easily installed on one or more personal computers. Moreover, the present invention can create, accept and track database queries in the more commonly used protocols, such as the Internet Protocol and translates them into proprietary or legacy protocols in real or near-real time. As a result, the present invention allows the user to better manage his or her risk with respect to call or transaction processing.

Please amend paragraph [0006] as follows:

[0006] The present invention provides a method of processing a database query between one or more clients and one or more databases. The database query, which is formatted using a first protocol, is received from one of the clients. One of the databases is selected to process the database query and the database query is translated from the first protocol to a second protocol. The translated database query is then sent to the selected database for processing. Thereafter, a response to the database query, which is formatted using the second protocol, is received from the selected database and the response is translated from the second protocol to the first protocol. The present invention then determines which of the clients sent the database query and sends the translated response to the client that sent the database query. This method can be implemented as a computer program embodied on a computer readable medium.

Please amend paragraph [0008] as follows:

[0008] The computer and the server/router module of system described above can be implemented using a first computer communicably coupled to the one or more clients, a second computer communicably to the one or more network servers, a server module resident on the first computer, and a router module resident on the second computer. The server module receives the database query, which is formatted using a first protocol, from one of the clients, sends the database query to the second computer, and sends a translated response to the client that sent the database query. The router module selects one of the network servers and one of the databases to process the database query, sends the database query to the selected network server, determines which of the clients sent the database query, and sends the translated response to the first computer[,].

Please amend paragraph [0013] as follows:

[0013] Referring to FIGURE 1, a block diagram of a network 100 in accordance with one embodiment of the present invention is shown. The network 100 includes one or more clients 102 communicably coupled to a server/router 104. The server/router 104 is communicably coupled to a[n] [s]Signaling [s]System 7 ("SS7") network 108 via SS7 server 106, one or more legacy networks 112 via legacy servers [I] 110, and one or more financial networks 116 via bank server 114. One or more financial institutions 118 may also be communicably coupled to the server/router 104. The clients 102 are computers or other devices that submit database queries for processing via the server/router 104. The functions of the server/router 104 can be split on separate computers or processing devices. The servers 106, 110 and 114 function as gateways between the network where

the server/router 104 is resident and the networks 108, 112 and 116 where the databases are resident.

Please amend paragraph [0020] as follows:

[0020] FIGURE 2B shows an implementation of the present invention wherein the functionality of the client 102 (FIGURE 1), server/router 104 (FIGURE 1) and server 106, 110 or 114 (FIGURE 1) are separated and placed on multiple devices 212, 214, 216 and 218, The client 212, which may be a computer, includes the client module. The server 214, which may be a computer, includes the server module. The router 216, which may be a computer, includes the routing module. The network server 218, which may be a computer, includes one [O]or [M]more of the network interface modules. The contrast between FIGURES 2A and 2B demonstrates the versatility of the present invention.

Please amend paragraph [0022] as follows:

[0022] The destination SS7 database processes the SS7 Query and creates a SS7 Response in block 318. Those skilled in the art will recognize[d] that the actual processing of the SS7 Query may be accomplished with a database interface application or other type of management software to control and handle queries to the database. The SS7 Response is then sent from the SS7 destination database to the SS7 network server 218 in block 320 where the SS7 network server 218 translates the SS7 Response (a database response in a SS7 format - the second protocol) into an IP Response (a database response in an IP format - the first protocol) in block 322. The IP Response is sent from the SS7 network server 218 to the router 216 in block 324 where the router 216 determines the client destination for the IP Response in block 326. The IP Response is

then sent from the router 216 to the server 214 in block 328 and then from the server 214 to the client 212 in block 330. The client 212 can then either disconnect from the server 214 in block 332 or repeat the process previously described for new database queries. Once disconnected, the process ends in block 334. Those skilled in the art will recognize that the process described in FIGURE 3 is not limited to the use of the IP and SS7 protocols.

Please amend paragraph [0023] as follows:

[0023] Now referring to FIGURES 4A, 4B and 4C, schematic diagrams of a more detailed generic translation method in accordance with one embodiment of the present invention are shown. The process starts in block 400. The user logs into the client module in block 402 and the client module connects to the server module in block 404. The server module validates the user and/or client module in block 406. If the validation is not successful, as determined in decision block 408, the server module denies access to the user and/or client module in block [4 10] 410 and the process loops back to allow the user and/or client module to try again in block 402 or 404.

Please amend paragraph [0024] as follows:

[0024] If, however, the validation is successful, as determined in decision block [409] 408, the user submits a query, which is formatted in a first protocol, to the server module via the client module in block 412. The query is then sent from the server module to the router module in block 414 where the router module determines the destination database and network interface module in block 416. If the destination address (database) and network interface module are not found, as determined in decision block 418, an error

message is returned to the client module via the server module in block 420 and the process loops back to allow the user to submit another query in block 412.

Please amend paragraph [0026] as follows:

[0026] The destination database processes the network query and create[d]s a network response in block 436. Those skilled in the art will recognize[d] that the actual processing of the network query may be accomplished with a database interface application or other type of management software to control and handle queries to the database. The network response is then sent from the destination address (database) to the network interface module in block 438 where the network interface module correlates the network response with the corresponding network query in block 440. The network interface module translates the network response (a database response in the second protocol format) into a client response (a database response in the first protocol format) in block 442. The client response is sent from the network interface module to the router module in block 444 where the router module correlates the client response with the query (determines the proper client module) in block 446. The client response is then sent from the router module to the server module in block 448 and then from the server module to the client module in block 450. The query timer is also turned off in block 450. If there are not other queries, as determined by decision block 452, the user logs out of the client module in block 454 and the process ends in block 456. If, however, there is another query, as determined in decision block 452, the process loops back to receive another query from the user in block 412.

Please amend paragraph [0027] as follows:

[0027] After the query has been sent by the server module in block 422, the query timer monitors [where] whether a response to the query has been received. If the timer has not expired, as determined in decision block 462, the query timer continues to wait for a response. If, however, the timer has expired, as determined in decision block 462, the server module sends its time out response to the client module in block 464[,]. As previously described, the client module, server module, router module and network interface module can be separated or combined in any configuration that fits the application of the present invention.

Please amend the Abstract as follows:

[0031] The present invention provides a system and method for processing a database query between one or more clients and one or more databases. The database query, which is formatted using a fast protocol, is received from one of the clients. One of the databases is selected to process the database query and the database query is translated from the first protocol to a second protocol. The translated database query is then sent to the selected database for processing. A response to the database query, which is formatted using the second protocol, is received from the selected database and the response is translated from the second protocol to the first protocol. The present invention determines which of the clients sent the database query and sends the translated response to the client that sent the database query. This method can be implemented as a computer program embodied on a computer readable medium.